

Hardware Requirements

Bedroom To Club Rig

DjMirror Pulse is not a lightweight slideshow app: before you switch on a single feature it is already running a real-time audio-analysis thread — a continuous 2048-point FFT (~46 ms window) plus BPM, key, energy and structure detection — and rendering GPU-accelerated shader...

DjMirror Pulse User Manual Series

Source: docs_user_manuals\SYSTEM_REQUIREMENTS_GUIDE.md

Revision 2026.10 • Release 1.0.743

Table of Contents

Chapter 1: Start Here

Chapter 2: Computer: Minimum And Recommended

Chapter 3: Graphics Card: The Real Bottleneck

Chapter 4: Audio Interface

Chapter 5: USB-DMX Controller

Chapter 6: Requirements Added By Optional Features

Chapter 7: Size Your Setup To Your Show

Chapter 8: The Android App

Chapter 9: Faq

The desktop application runs natively on **Windows 10/11 (64-bit)** and **macOS (Intel and Apple Silicon)**. The **Android app is a remote control** for a running desktop installation — it does not run the audio-analysis or rendering engine itself, so it does not need this hardware. The figures below are **engineering sizing guidance**, not certified minimums or guaranteed performance: your result depends on the GPU, output resolution, effects, media and connected hardware, so rehearse your actual setup before a show.

Chapter 1: Start Here

If you only remember four things, remember these.

#	Rule	Why
1	GPU first	The graphics card is the real bottleneck, not the CPU. Every layer, shader and particle renders through it.
2	Real audio input	A laptop microphone won't cut it. You need a line/loopback input carrying your mixer or player signal.
3	Stable USB	Keep every USB-DMX interface and controller on dedicated, stable ports — never behind a flaky hub during a live set.
4	16 GB RAM	Comfortable for a full live rig with video backgrounds, DMX, MIDI and NDI running together.

The app also keeps DMX, MIDI, HID and the embedded HTTP/WebSocket/MCP servers on their own threads. A weak CPU will not crash it, but it shows up as analysis jitter and MIDI-to-visual latency creeping above the sub-millisecond target. Related reading: [Audio Setup & Beat Sync](#), [MIDI & HID Controllers](#).

Chapter 2: Computer: Minimum And Recommended

	Minimum	Recommended (live performance)
Operating system	Windows 10 64-bit (21H2 or later) / macOS 11	Windows 11 64-bit / macOS 13+ (Apple Silicon)
CPU	Quad-core, 2.5 GHz (Intel i5 8th gen+ / Ryzen 3 / Apple M1)	6+ cores, 3.2 GHz+ (Intel i7 10th gen+ / Ryzen 5–7 / Apple M2–M3 Pro)
RAM	8 GB	16 GB (32 GB with local AI models or large media libraries)
Storage	2 GB free, HDD acceptable	5 GB+ free on SSD/NVMe (video backgrounds and screen recordings add up fast)
Display	1280×720 @ 60 Hz	1920×1080 @ 60 Hz or higher, dual monitor (control panel + full-screen output)
Network	Needed only by the features that use it — local visuals, audio, DMX and MIDI work fully offline	Wired Gigabit Ethernet for NDI and multi-screen; a network adapter is also needed for the mobile remote, the HTTP/MCP API and cloud services
Instruction set	SSE4.2 (x86-64 baseline)	AVX2 is required only for the optional local AI features (offline image generation via llama.cpp; not included in the Mac App Store build)

Most Intel/AMD chips from the last six or seven years already support AVX2. The recommended column is not required to run the app — it is the column that keeps multi-layer shows smooth for hours.

Chapter 3: Graphics Card: The Real Bottleneck

Every visual layer — waveforms, spectrum and XY visualizers, the vinyl disc, particles, virtual lighting, GLSL shaders and marquee text — is rendered through Qt's hardware-accelerated engine (OpenGL/Vulkan on Windows, Metal on macOS). Integrated graphics can comfortably handle one background plus one visualizer; the moment you stack several audio-reactive layers, a particle system and a video background at once — a typical live-performance setup — a discrete GPU is what keeps the frame rate smooth.

	Minimum	Recommended
Windows	DirectX 11 / OpenGL 3.3, 2 GB VRAM (e.g. Intel UHD 620/630 integrated)	DirectX 12 / Vulkan discrete GPU, 4–8 GB VRAM — NVIDIA GTX 1660 / RTX 3050+ or AMD RX 6600+
macOS	Metal-capable integrated GPU (Apple M1 or Intel Iris Plus, 2018+)	Apple Silicon Pro/Max-tier GPU, or a discrete AMD Radeon Pro on eligible Intel Macs
4K video backgrounds	Not guaranteed smooth	6 GB+ VRAM with hardware HEVC/H.264 decode

If you are choosing between upgrading the CPU or the GPU on a tight budget, **upgrade the GPU** — it is what carries the visual workload on every single set. Do not assume a particular card guarantees a frame rate: start at the output resolution you will actually use, then add layers one at a time while watching for dropped frames. More detail: [Visualizers](#), [Layers](#), [Shaders & FX](#).

Chapter 4: Audio Interface

The audio-analysis engine needs a real, continuous signal to analyze — BPM, key, energy, structure and event detection all depend on it. Your built-in laptop microphone is not a viable source for a live set: you need a proper **line-level input** carrying the signal from your mixer, controller or media player.

Recommended input paths

1. An external USB audio interface (Focusrite Scarlett, Behringer UMC, PreSonus AudioBox, RME Babyface, or similar) fed by your mixer's booth/master out.
2. A DJ controller/mixer with a built-in class-compliant USB audio output (most Pioneer, Denon, Native Instruments and Numark units qualify).
3. A software loopback device capturing your player's output directly — Windows: a virtual audio cable/WASAPI loopback; macOS: BlackHole or Rogue Amoeba Loopback.

What to look for

- Class-compliant USB audio on Windows (WDM/WASAPI) or Core Audio on macOS — no exotic proprietary drivers.
- Low, stable input latency: the analysis engine reacts within a ~46 ms FFT window, so a jittery driver undermines beat/drop accuracy.
- A dedicated line/aux input separate from your monitoring output, so you are not fighting for the same channel.
- A level that is neither silent nor clipping. A built-in microphone can be used for a quick test, but room noise gives inconsistent results.

Step-by-step device selection and the silence detector: [Audio Setup & Beat Sync](#).

Chapter 5: USB-DMX Controller

DjMirror Pulse auto-detects supported USB-DMX interfaces by port and VID/PID — plug one in and it is ready, with no manual protocol configuration. Every interface exposes one or more universes of 512 channels each, routed to your fixtures by **outputPort** + **universe**. Several interfaces can run at once, so your lighting rig can grow with you. Windows sometimes reports an FTDI interface only as "USB Serial Port"; in that case select the controller type by hand or use **Detect Protocol**.

Interface	Universes	Best for
Enttec Open DMX USB	1	Entry-level rig, budget-friendly first controller
Enttec Pro DMX USB	1	Reliable single-universe professional interface
Enttec Pro DMX USB Mk2	1 (port 1; the second port needs an Enttec developer key)	Mk2 owners — see the note below
DMXKing ultraDMX Micro / Pro	1	Compact, auto-detected single-universe setups
DMXKing ultraDMX2 Pro	2 on one cable	Two-universe rigs without doubling your USB ports
Eurolite USB-DMX512 PRO / PRO MK2 (DMX Pro v2)	1	Widely available in the EU lighting market
SoundSwitch Micro DMX Interface	1	Musicians/producers already in the SoundSwitch ecosystem — use the generic FTDI / Enttec Open DMX class and select it manually if it is not auto-detected
FreeStyler USB DMX / generic FTDI	1	Budget and DIY-friendly interfaces
DMXIS	1	Combined controller + interface hardware

Two universes: the published requirements page lists the Enttec Pro Mk2 as a two-universe interface. In the current build only **port 1** is driven (the secondary port labels require a vendor key), so for two 512-channel universes use a **DMXKing ultraDMX2 Pro** or two single-universe interfaces on separate USB ports.

Just getting started? A single Enttec Open DMX USB or a DMXKing ultraDMX Micro is plenty to drive a handful of PARs and a moving head for your first live rig — both are auto-detected the moment you plug them in.

Running a full show? Go for a DMXKing ultraDMX2 Pro, or connect two single-universe interfaces on separate USB ports. Keep each USB-DMX interface on a stable, dedicated port — avoid re-enumeration through a cheap hub mid-set, and close QLC+, FreeStyler or vendor utilities, since only one application can hold the serial port at a time. Patching and diagnostics: [DMX Rig Configuration](#).

Chapter 6: Requirements Added By Optional Features

These are **additive** to the tables above — size only the rows that apply to your setup.

Feature	Additional requirement
Video backgrounds (1080p)	Discrete GPU with hardware video decode; use at least the recommended CPU/GPU tier
Video backgrounds (4K)	6 GB+ VRAM with hardware HEVC/H.264 decode; recommended CPU tier
Several DMX universes	One free, stable USB port per interface; no significant CPU/RAM overhead per universe
NDI output	Wired Gigabit Ethernet (not Wi-Fi); NDI runtime installed; extra CPU headroom — see NDI & Spout Output
Screen recording (MP4)	Free disk space sized to session length (a 30-minute recording at the default 20 fps can reach several hundred MB); SSD avoids dropped frames — see Recording & Snapshots
Local AI (llama.cpp; not included in the Mac App Store build)	AVX2 CPU; 16 GB+ RAM for small quantized models, 32 GB+ for larger ones; model files are user-supplied (typically 2–8 GB each)
Cloud AI / stock media search	Internet connection only — see Media Search & AI

Chapter 7: Size Your Setup To Your Show

Tier	Typical usage	What it drives
Basic	Single visual layer (images + waveform), no external hardware	The minimum spec is enough
Standard live	Multiple layers, particles/shaders, a MIDI controller, one USB-DMX interface	The recommended spec
Advanced production	1080p/4K video backgrounds, several DMX universes across multiple interfaces, NDI output, screen recording and/or local AI running together	High-end recommended spec + wired Gigabit network + SSD/NVMe storage

For NDI or multi-screen use, run the connection over **wired Gigabit Ethernet** rather than Wi-Fi: NDI is bandwidth-intensive, and a wired connection on the same LAN segment as your receiver avoids dropped frames. Test the actual venue network and receiver setup — [Multi-Screen & Client-Server](#).

Chapter 8: The Android App

DjMirrorPulse for Android is a **remote control for your desktop install** (Windows or macOS). It does not run the audio-analysis or rendering engine itself, so it does not need the hardware in this guide. What it does need:

- DjMirror Pulse running on your PC/Mac, with the local server enabled in **Settings** → **Mobile app** (`"server": true` or `--server`) and an active license.
- Your phone and computer on the **same network**.
- A free phone slot for your edition: Trial/Standard 1, Professional 2, Enterprise 5.
- No pairing hardware: pair once with the one-use QR code shown in **Settings** → **Mobile app** → **Pair a phone** (valid 180 s), or type the short code on the phone if it has no camera. The desktop listens on its LAN address; default ports are `8080` (HTTP) and `8081` (WebSocket).

From your phone you can switch styles, toggle the visual layers, jump between playlists, and run the full DMX Scene Player — including blackout, fog/haze/strobe bursts and live per-channel sliders. Full workflow: [Mobile Remote](#).

Chapter 9: Faq

Can I run DjMirror Pulse on a laptop with only integrated graphics? Yes, for a simple style with one background and one visualizer — that is the minimum tier. For a full multi-layer live set with particles and a video background, plan on a discrete GPU.

Do I need a dedicated audio interface if I already use a DJ controller? If your controller exposes a class-compliant USB audio output, that is all you need — route its master/booth channel into DjMirror Pulse as the analysis input. A separate interface is only necessary if your setup does not already provide one.

Which USB-DMX controller should I buy first? An Enttec Open DMX USB or a DMXKing ultraDMX Micro covers most first rigs. If you already know you will need two universes, go straight for a DMXKing ultraDMX2 Pro.

Is the Mac App Store version different? Yes. Because of the store's sandbox rules, the Mac App Store build leaves out NDI, local AI inference (llama.cpp), Hugging Face downloads, Ableton Link and Traktor integration. NDI and local AI are the heaviest optional features, so that build needs less hardware for the same show. Everything else follows the guidance above. Your license edition (Standard, Professional or Enterprise) is decided by your activation key, not by the build — see [License & Activation](#).

How do I know if my computer is enough? Rehearse your intended styles, media, resolution and connected hardware together, then check for stutter, dropped frames or audio dropouts before taking it to a venue — see [Troubleshooting](#).

Can I test before buying? Download the free trial (14 days, Standard tier) and measure on your own rig — see [License & Activation](#).